MPW Conversion

Development platform for local station system software

Jan 13, 1992

During a ten day period around New Year's Day 1992, the local station system software was ported to the MPW development system used on the Macintosh. This brief note describes something of how the process went. Reasons for the change from the Octal tools on the Vax used for many years are these:

1. Octal software is no longer actively supported. The 68020 assembler seemed to be unable to generate the 32-bit PC-relative addressing mode that is needed to get an address of a routine or table in the system code in a position-independent way in a large (> 32K) program.

2. The Pascal compiler pre-dates the floating point co-processor chip and cannot generate the instructions for type Real support. Local applications that need such instructions were developed with MPW Pascal.

3. The size of the system code (60K) was stretching to the limit a model of programming conventions that would theoretically break at 64K, as it attempted to maintain the limits of 32K branches of the 68000 CPU. Since all local stations now run on 68020 CPU's, this restriction can be eliminated.

4. The Macintosh is a conveniently accessible platform if one has a Macintosh on his desk. It is actively supported by Apple Computer for its own software development and for Macintosh developers. The cost is $525 for a single user, with support for both C and Pascal.

5. The most recent version of MPW, version 3.2, allows support of "32-bit everything" including the ability to generate a code resource >32K bytes. The local station system code is linked as a single code block segment.

6. The MPW assembler supports structured assembly syntax through an extensive set of macros. Many other assemblers available today do not include support for structured syntax (IF-THEN-ELSE, REPEAT-UNTIL, etc). The local station system software uses this syntax heavily. Elimination of that syntax from the source code is inconceivable.

The first job to be handled was to find a way to automatically change the structured assembly syntax from the format Octal supported (and Motorola in their original EXORmacs development system) into the format used by MPW. The MPW shell supports Unix-like (but not fully Unix-compatible) commands for editing that include "regular expressions" for indicating the text to match for a Replace command. A set of 20 Replace commands were designed for use via a command file to do the structured syntax conversion. Without this automatic aid, the job

Each of the approximately 100 source files of 31K lines of assembly code was passed through the automatic conversion first and boiler-plate header lines inserted to invoke the structured syntax macro support and the system constants "include" file. Then each file was checked for details that had to be changed manually. This included identifying code sections individually with the PROC directive and data structure templates using the RECORD directive. There were also some pathological cases converted incorrectly by the automatic process.

Then an assembly was attempted. Sometimes no errors resulted. Sometimes up to 10 pages of errors were listed on the screen. In the latter case, usually a few error corrections eliminated many others. Eventually, all errors were gone, and it was time to move on to the next one. After the first few dozen files were converted, things began to go faster.

After all source files were converted and assembled, the next step was to link them into a system. At this point MPW 3.2 was needed to get the "-model far" option that permits building a large CODE resource. After correcting the errors in undefined names, a 60K CODE resource was in hand.

The next step was to use the Hex tool to translate into S-records to be used for downloading via the serial port into a VME local station. This tool was written by someone from England long before, and it was limited to CODE resources of 32K bytes. Inspection of the Pascal source code revealed that the buffer was only 32K in size. The size was changed to 100000 (decimal), and the tool was re-compiled with the "-model far" option used for both the compiler and linker commands in the "make" file. It worked.

The S-records were downloaded into local station 0576, and after a reset, the system came up running. But a bug was found that produced a "3" error (internal inconsistency) on the memory dump page. Upon correction, the system ran apparently ok. In the next days, a few more errors were found and corrected, including the macro file FlowCtlMacs that supports the structured syntax.